REMARKS

Applicant appreciates the Office's consideration of the Response filed August 5, 2005. In an Advisory Action dated September 16, 2005, the Office stated that the August 5, 2005 Response will not be entered. Applicant does not understand why the Office has made this decision. The Response filed on August 5th sets forth arguments that should be part of the record should the Applicant decide to go forward with an Appeal; moreover, the filed Response fully complies with 37 CFR 1.116. Therefore, the Office is respectfully requested to indicate in response hereto that the Response of August 5, 2005 will be entered upon the timely filing of an Appeal Brief, should one be necessary. Applicant hopes such a necessity will be avoided by the filing of this additional Response.

In an effort to place the present Application in condition for allowance, Applicant submits this Response. The Response includes amendments to independent claims 28, 49, 52, 56 and 60. The claims include limitations from claims dependent claims canceled without prejudice or disclaimer in this Response. The canceled claims include 29-30, 50-51, 53-54, 58-59 and 62-63. Claims 28, 48-49, 52, 55-57, 60-61 and 64 remain pending in the present Application.

This Response complies with 37 CFR 1.116 and therefore should be entered by the Office. In particular, this Response does not raise new issues that require consideration by the Office; the limitations added to the claims were previously considered by the office in various dependent claims canceled without prejudice or disclaimer hereby.

Applicant respectfully requests reconsideration and allowance of the subject application in view of the amendments and comments provided in the foregoing. Claims 28, 48-49, 52, 55-57, 60-61 and 64 are pending in the application.

The Applicant appreciates the time the Examiner afforded to the Applicant's representative during a recent conversation related to the status of the present Application. The Examiner and the Applicant's representative discussed the status of the claims during that conversation. More specifically, the various limitations now set forth in the independent claims as amended were discussed in relation to the patent cited in the sole rejection on record. The Examiner indicated that he may view such amendments favorably if submitted as part of a Response.

Claim Rejection under 35 U.S.C. § 102

Claims 28-30 and 48-64 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,098,093 to Bayeh et al. (hereinafter, "*Bayeh*"). Applicant respectfully traverses the rejection.

Amended **Claim 28** defines a stateless distributed computer system, comprising:

> a network having one or more network components to route requests from a first endpoint device to a second endpoint device and to route replies from the second endpoint device back to the first endpoint device, wherein at least one reply contains state information pertaining to the second endpoint device; and
>
> the network being configured to maintain the state information and to reassociate the state information with a subsequent request from the first endpoint device to the second endpoint device, and *wherein multiple network components continually route the state information amongst themselves to preserve the state information.* (Emphasis added)

As recited in claim 28, the claimed stateless distributed computer system includes a network between two endpoints and the network is configured to maintain the state information, rather than the state information being kept at either of the two endpoints. In particular, claim 28 recites that multiple network components continually route the state information amongst themselves to preserve the state information.

As described in one exemplary implementation in the subject application, with reference to Fig. 8 (reproduced below) and accompanying text beginning on page 17, a network system 800 has a first endpoint device 802 and a second endpoint device 804 interconnected via a network 806. The network 806 includes one or more specially configured computing devices whose task is to route messages between the endpoint computing devices 802 and 804. The network computing devices may include routers, hubs, relays, repeaters, satellite uplinks and downlinks, RF transceivers, and the like.

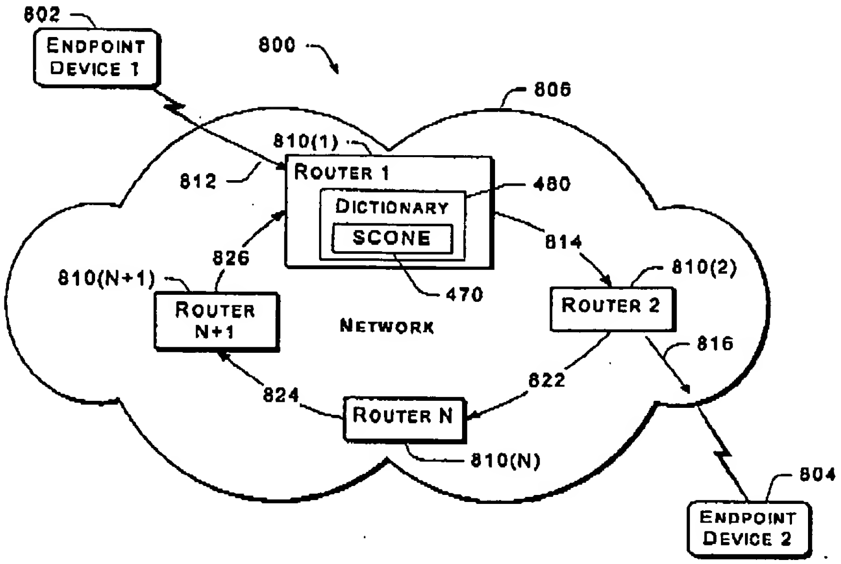


Fig. 8

A message may be routed, for example, from the first endpoint 802 to the second endpoint 804 through routers 810(1) and 810(2) along path segments 812, 814, and 816. Suppose the second endpoint 804 responds to the request by returning a reply packet that contains a "state-caching object for a network element" or "SCONE" 470. The reply packet may be routed back to the first endpoint 802 via the same or different path through the network 806.

Rather than caching the SCONE 470 on the first endpoint 802 or second endpoint 804, the network 806 keeps the SCONE 470 on behalf of the two endpoint devices 802 and 804.

According to one implementation, a network component copies the SCONE 470 from the reply packet and stores it. This is represented in Fig. 8 by the SCONE 470 being stored in router 810(1) in a dictionary 480 by service ID. If the first endpoint device 802 subsequently sends another request to the second endpoint device 804, the router 810(1) notes the reuse of the service ID and reattaches the SCONE 470 to the packet to return the state information to the second endpoint device 804. If no subsequent request is made, the SCONE 470 remains on the router 810(1) until it expires and is removed from memory.

According to a second implementation, the SCONE 470 is not kept at one router, but instead is continuously routed among various network components indefinitely or until timeout. In this example, the SCONE 470 may be circulated among four routers 810(1), 810(2), 810(N), and 810(N+1), as represented by path segments 814, 822, 824, and 826. If a subsequent connection between the first and second endpoint devices is made, first router 810(1) to transport the message issues a distributed query to the other routers 810(2), 810(N), and 810(N+1) to locate the matching SCONE 470 if any. The SCONE 470 is subsequently reassociated with a request and returned to the second endpoint 804 to restore state information.

To summarize the above exemplary embodiments of the present invention, the network 806 is the communication medium for the first endpoint device 802 and the second endpoint device 804. When the first endpoint device 802 sends a communication/request to the second endpoint device 804, the network 806 facilitates routing that communication in an appropriate manner. Similarly, the network 806 allows the second endpoint device 804 to send communications/responses to the first endpoint device 802. The endpoint devices 802 and 804 may be various computing devices (e.g., clients, servers, server cluster/farm).

Prior art networks are fundamentally different from the network 806 of the exemplary embodiments of the present invention. Such prior art networks are only responsible for routing communications to and from the various computing devices that are connected thereto. The network 806 offers the additional and advantageous capabilities discussed above.

Turning now to the *Bayeh,* the relied upon document fails to disclose the system of claim 28. *Bayeh* discloses a system for maintaining sessions in a clustered server environment. The sessions are maintained as "servlets", which are small executable code objects used in Java-based products. In the Background section, *Bayeh* noted that one such product, the Java Web Server Toolkit from Sun Microsystems, only described a session tracking facility for a single Web server. (*Bayeh*, col. 4, lines 61-64). Hence, the goal of *Bayeh* was to extend session services to a clustered server environment. (*Bayeh,* col. 8, lines 59-66).

Bayeh describes a clustered server environment where multiple Web servers 60, 62, and 64 are arranged behind a load-balancing host 59 to receive and respond to incoming client requests 100, 101, and 102. (*Bayeh*, Fig. 3, col. 8, lines 42-58). A servlet engine 70, 72, and 74 is provided at each Web server. (*Bayeh*, col. 8, line 64 to col. 9, line 6). *Bayeh* describes that session information used to respond to client requests can be maintained by the servlet objects, and kept in a session pool of a session server for subsequent transactions. All of this occurs at the server cluster behind the load-balancing host 59.

To describe further, *Bayeh* teaches that the session information is stored in a Web server 60 (session server) that includes the servlet engine 70. According to Bayeh, this Web server 60, which is designated to store the session information, does not accept and/or respond to client requests. (*Bayeh*, col. 9, lines 31-38). As a matter of fact, when a given Web server is acting as the session server, the load balancer 59 ensures it never receives client requests.

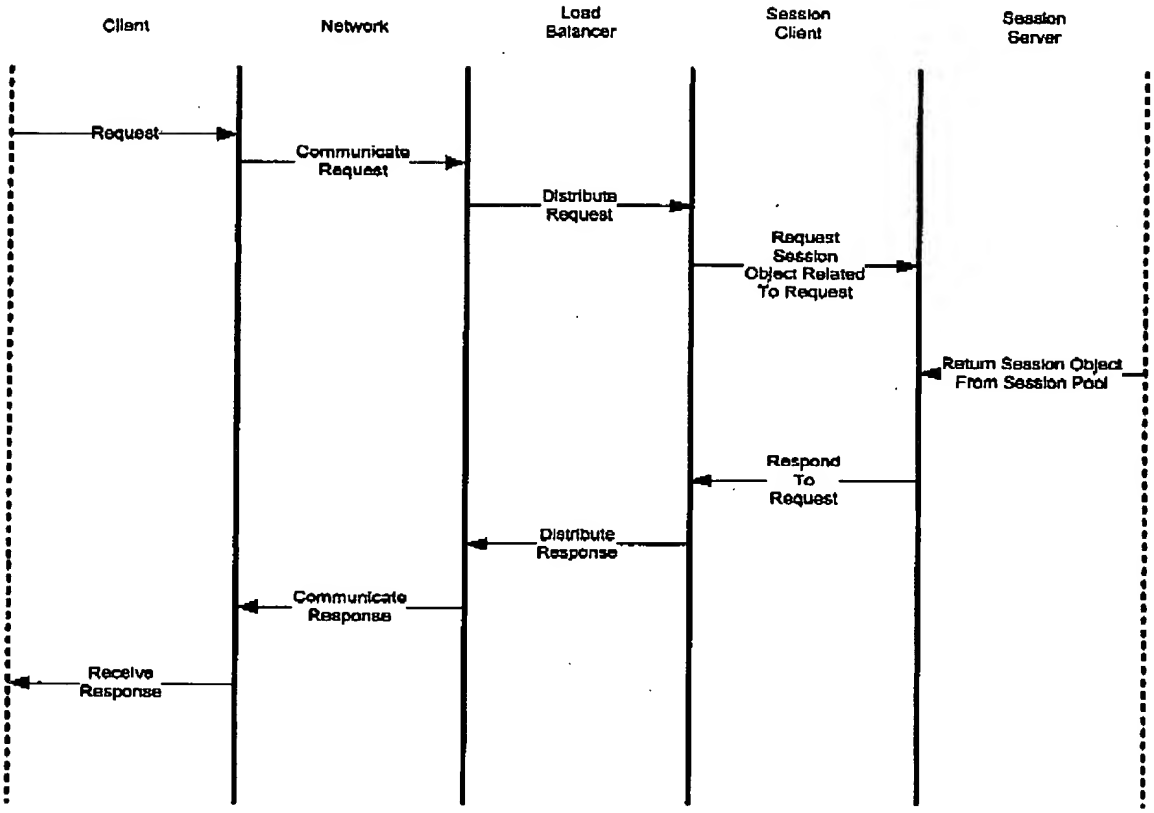
Figs. 4A-4C of *Bayeh* illustrate a process that occurs when a request is received by the Web server cluster illustrated in Fig. 3. When a client request/communication is received, the load balancer 59 makes a determination as to which server (60, 62, or 64) in the Web server cluster should receive the client request. As was discussed above, the server that is designated as the session server is configured as a non-receiving source for incoming client requests/communications. Accordingly, the load balancer 59 will forward the client request to Web server 62 or 64.

The server receiving the client request uses a servlet engine to invoke a method that is defined to retrieve session information related to the client request. (*Bayeh*, col. 11, lines 20-27). The invoked method allows the server to communicate with the session server to retrieve the session object from the session pool held by the session server. (*Bayeh*, col. 11, lines 43-46).

Several processes occur at the session server once the server that received the client request begins communicating with the session server. This Response will discuss only those actions asserted by the Office as being relevant to the present invention.

When a valid request for session information is received by the session server, a servlet engine of the session server retrieves the appropriate session object from the session pool and returns it to the server requesting the information. (*Bayeh*, col. 12, lines 22-28; col. 13, lines 7-8). Then, the sever that received the client request may respond to the client request issued by the load balancer 59.

To aid the Office's understanding of the invention according to *Bayeh*, the Applicant provides the following diagram of interactions that occur once the load balancer 59 receives a client request from a requesting client.



Bayeh's architecture merely describes maintaining state information at a server cluster. The server cluster includes one server 60 that behaves as the session server and holds all of the session objects in a session pool, and one or more servers (62, 64) that behave as session clients for handling client requests distributed by a load balancer 59. The distribution in the architecture is shown in the diagram provided on the foregoing page.

Nowhere does *Bayeh* ever show or consider a network between the client(s) and the server cluster (e.g., a network between the client(s) and the load-balancing host in Fig. 3), where the network itself maintains the state information. Furthermore, *Bayeh* does not teach or suggest multiple network components continually route the state information amongst themselves to preserve the state information. (See claim 28.) Instead, the network according to the *Bayeh* architecture, which is illustrated in the indicated diagram, is merely a medium for conveying information.

In addition, *Bayeh* is entirely silent as to the "stateless distributed computer system" of claim 28, as *Bayeh* does not discuss or disclose "a network having one or more network components to route requests from a first endpoint device to a second endpoint device" where "the network [is] configured to maintain the state information and to reassociate the state information with a subsequent request from the first endpoint device to the second endpoint device" as required by claim 28.

In the current Office Action, the Office states "the two clients are part of the network, thus the information is already part of the network." It is unclear what the Office asserting with this statement. If the Office is saying the network between the session server and the session client maintains the session objects, the Applicant respectfully disagrees. Nowhere does *Bayeh* teach or suggest this concept. Instead, *Bayeh* is clear that the session objects are held in a session sever of the server cluster.

The session server is not the same as the "multiple network components" described in claim 28. It must be if *Bayeh* was correctly relied upon by the Office. As is set forth in claim 28, "multiple network components continually route the state information amongst themselves to preserve the state information." Session objects are held in the session server taught by *Bayeh*, but nothing in the relied upon patent document teaches or suggests that the session server reassociates these session objects with "a subsequent request from the first endpoint device to the second endpoint device," as does the network set forth in claim 28. Instead, as is shown in the diagram on page 17 of this Response, the session client that requests a given session object and retrieves the same from the session server is the entity in *Bayeh* that handles the reassociation process. However, the session client is unable to store session objects, since the session server of the server cluster has this sole responsibility. (*Bayeh*, col. 9, lines 26-38). Therefore, both the session client and the session server are unable to operate in the manner the "network" of claim 28 functions.

With the session client and server eliminated as candidates that teach the "network" set forth in claim 28, the only remaining device taught in *Bayeh* that processes client requests is the load balancing host 59. *Bayeh* indicates the load balancing host 59 operates in a known manner. (*Bayeh*, col. 8, lines 49-58). Therefore, the load balancing host 59 simply routes client requests to servers in the server cluster based on an amount Web traffic being handled by the various servers in the cluster. Therefore, the load balancing host 59 does not function in the same manner as the "network" recited in claim 28.

For the reasons stated above, claim 28 is allowable over *Bayeh*. Applicant respectfully requests that the § 102 rejection be withdrawn.

Dependent claim 48 depends from claim 28 and is allowable by virtue of this dependency. Moreover, the claim recites features that, when taken together with those of claim 28, define systems not disclosed by *Bayeh*.

Therefore, the implementation set forth in claim 48 is not described in *Bayeh*.

Claims 48-49, 52, 55-57, 60-61 and 64

Independent **claims 49, 52, 56,** and **60** set forth subject matter similar to that discussed in conjunction with claim **28**. Accordingly, these claims and those claims dependent thereon are allowable over *Beyeh*.

Claim 49 recites computer-executable instructions at least capable of directing a system to "continually route the state information among multiple network components to preserve the state information." *Beyeh* simply does not teach or suggest the indicated features of claim 49.

Claim 52 recites "network means comprising means for maintaining the state information within the network means and for reassociating the state information with a subsequent request from the client to the server, and *means for continually routing the state information among network components to preserve the state information*." (Emphasis added.) *Beyeh* does not teach or suggest the features of this claim as well.

Claim 56 recites "maintaining the state information at the network by continually routing the state information among network components of the network to preserve the state information." *Beyeh* does not teach or suggest at least the indicated limitations of claim 56.

Claim 60 recites "maintaining the state information on the network while awaiting a subsequent request from the client to the server *by continually routing the state information among network components of the network to preserve the state information*." (Emphasis added.) *Beyeh* does not teach or suggest the features of this claim as well.

Conclusion

Claims 28, 48-49, 52, 55-57, 60-61 and 64 are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of the subject application. If any issue remains unresolved that would prevent allowance of this case, **the Examiner is requested to contact the undersigned attorney to resolve the issue**.

Respectfully Submitted,

Date: 10-25-2005

By: ____________________
Tim R. Wyckoff
Lee & Hayes, pllc
Reg. No. 46,175
206.315.4001 x110